Práctica 1: Problema del Bandido de k-brazos.

Alejandro López Cuéllar, Miguel Ángel Vera Frutos, Alejandro Belda Fernández

9 de marzo de 2025

1. Introducción

El bandido de k brazos, también conocido como bandido multibrazo (*K-bandit* en inglés), hace referencia al problema en el que una persona se encuentra frente a una serie de máquinas tragaperras. De este modo, el jugador deberá seleccionar con qué máquina jugar en cada momento, ya que las recompensas ofrecidas por cada una de las máquinas vienen dadas según una distribución de probabilidad propia de cada tragaperras. Además, partiendo de la base de que el jugador no posee ninguna información previa acerca de las recompensas que ofrecen dichas máquinas.

Se establece pues como objetivos para la práctica:

- Realizar el estudio de distintos métodos empleados para abordar el problema del bandido multibrazo.
- Implementar dichos métodos estudiados.
- Realizar experimentos con cada uno de los métodos empleados para ver su funcionamiento con distintos tipos de distribuciones de probabilidad.

Así pues, a lo largo de este documento se tratarán distintos puntos relacionados con dicha práctica.
En un primer lugar, en el apartado de **Desarrollo** se especificará la definición formal del problema; tras ello, se comentará los algoritmos referentes a cada método empleado en el apartado de **Algoritmos**. Una vez comentados los algoritmos empleados, en el apartado de **Evaluación/Experimentos** se mostrarán las pruebas realizadas con cada uno de los métodos y las distintas distribuciones de probabilidad.

2. Desarrollo

2.1. Descripción

El problema del **bandido multibrazo** es un problema clásico de aprendizaje por refuerzo. Se define formalmente de la siguiente manera:

- Existen k acciones diferentes a elegir en cada instante de tiempo t.
- Elegir una de las acciones $a_t \in A$ ofrece una recompensa r_t.
- Cada uno de los brazos k posee una distribución de recompensas que es desconocida para el usuario.
- Se establece como objetivo maximizar el valor de la recompensa acumulada que se obtiene: $\sum_t r_t$.
- La presencia de distribuciones de probabilidad desconocidas para el jugador presentan un problema, de tal modo que se debe elegir entre **explorar**, probando distintos brazos para comprobar sus recompensas y **explotar**, eligiendo el brazo k que mejor valor de recompensa esperada q_a posea.

2.2. Métodos utilizados

De este modo, los métodos empleados para trabajar en el problema del bandido multibrazo han sido los siguientes:

- Método ϵ greedy
- Métodos UCB:
  - UCB1
  - UCB2
- Métodos de Ascenso de Gradiente:
  - Softmax
  - Gradiente de preferencias

2.3. Distribuciones de probabilidad

A su vez, para trabajar con las distintas distribuciones de probabilidad se han implementado funciones para generar los brazos del bandido según la distribución normal, distribución binomial y distribución bernoulli.

3. Algoritmos

3.1. Algoritmos de los métodos usados

A continuación, se realizará la explicación de los algoritmos utilizados para abarcar el problema del bandido multibrazos.

3.1.1. ϵ-Greedy

Mediante el método épsilon-greedy se pretende realizar solucionar el problema que viene dado en el método greedy, el cual está centrado en la explotación.
Mediante el valor épsilon se permitirá realizar una exploración de los brazos pese a poder no ser lo que mayor recompensa esperada posean.

Así pues, el modo de proceder es el siguiente:

- Se genera un número aleatorio entre 0 y 1.
- Se compara con $\boldsymbol{\epsilon}$, si el valor de ϵ es mayor se selecciona de forma aleatoria uno de los brazos, excluyendo el que mayor recompensa esperada posee.
- Si el valor de $\boldsymbol{\epsilon}$ es menor que el número generado, por el contrario, se seleccionará el brazo óptimo.

De esta forma, se permite cierto balance entre explotación y exploración, evitando centrarse únicamente en uno de los brazos.

Algorithm 1 Selección de brazo con ε-Greedy

Require: $k > 0$ (número de brazos), $0 \leq \varepsilon \leq 1$ (probabilidad de exploración)
Require:
Ensure: Índice del brazo seleccionado
Generar un número aleatorio $r \sim U(0, 1)$
if $r < \varepsilon$ **then**
    $chosen\_arm \leftarrow$ seleccionar un brazo al azar en $\{0, \ldots, k-1\}$
else
    $chosen\_arm \leftarrow \arg\max(values)$
end if
return $chosen\_arm$

3.1.2. UCB1

Los métodos UCB basan su forma de actuar en el **límite superior de confianza**, este valor mide el potencial de los brazos mediante la suma de dos elementos. **Q(a)**, siendo el término que representa la parte de explotación; y **u(a)** siendo el término que representa la exploración.
El hecho de que **Q(a)** esté asociado a explotación y **u(a)** a explotación viene dado por el hecho de que en la suma de estos dos valores para la obtención de **ucb(a)**, cuando uno de ellos es alto, el otro es bajo. Por tanto, si un brazo se selecciona por ser el que mayor valor ucb posee, y a su vez dicho valor ucb es grande debido a (Q(a)) es causado porque la recompensa esperada en ese brazo es alta. Sin embargo, si se selecciona por su alto valor en **u(a)** será porque se pretende realizar exploración pese a que no tenga una gran recompensa esperada.

Así pues, el método **UCB1** procede de la siguiente manera:

- En primer lugar se realiza una primera pasada por todos lo brazos, de modo que si un brazo no ha sido seleccionado se devuelve inmediatamente.
- En las siguientes selecciones, se lleva a cabo el cálculo de **t** para contabilizar el número de acciones que se han realizado.
- Se obtiene el valor **ucb1** para cada uno de los brazos.
- Finalmente se selecciona el brazo con un mayor valor **ucb1**, devolviéndolo.

Algorithm 2 Selección de brazo con UCB1

Require: $k > 0$ (número de brazos)
Ensure: Índice del brazo seleccionado
for $arm = 0$ **hasta** $k - 1$ **do**
    if $counts[arm] = 0$ **then**
        return arm
    end if
end for
$t \leftarrow \sum counts$
for $arm = 0$ **hasta** $k - 1$ **do**
    $ucb1[arm] \leftarrow values[arm] + \sqrt{\frac{2 \log t}{counts[arm]}}$
end for
$chosen\_arm \leftarrow \arg\max(ucb1)$
return $chosen\_arm$

3.1.3. UCB2

Al igual que el método UCB1, este método se basa en el **límite superior de confianza** (Upper Confident Bound). Sin embargo, a diferencia de **UCB1**, en este método se introducen las siguientes variables:

- $\boldsymbol{\alpha}$, un parámetro que sirve para ajustar el equilibrio entre explotación y exploración del método.
- k_a, variable que lleva el recuento de las épocas de una acción.
- $\boldsymbol{\tau}$, variable que determina el número de veces que se seleccionará un brazo determinado dentro de una época.

El proceso pues, es el siguiente:

- Se hace una pasada inicial por cada uno de los brazos, de tal forma que si uno de los brazos todavía no ha sido seleccionado, se devuelve de forma inmediata.
- Se comprueba si al brazo que está seleccionado le queda por ejecutar la acción alguna vez dentro de la época.
- Se obtiene el número de acciones que se han tomado, **t**.
- Para cada uno de los brazos, se calcula el valor $\boldsymbol{\tau}$
- Se calcula el valor **ucb2** para cada uno de los brazos.
- Se selecciona el brazo con un mayor valor **ucb**.
- Se calculan los valores $\tau(k_a)$ y $\tau(k_a + 1)$ y se restan. Para posteriormente obtener el número de veces que se tomará la acción del brazo seleccionado en la siguiente época.

Algorithm 3 Selección de brazo con UCB2

Require: $k > 0$ (número de brazos), $0 < \alpha < 1$ (parámetro de exploración)
Ensure: Índice del brazo seleccionado
if $brazo\_seleccionado \neq$ `None` y $rondas\_restantes > 0$ **then**
    $rondas\_restantes \leftarrow rondas\_restantes - 1$
    return $brazo\_seleccionado$
end if
for cada $brazo$ en $\{0, \ldots, k-1\}$ **do**
    if $conteos[brazo] = 0$ **then**
        $brazo\_seleccionado \leftarrow brazo$
        $\tau(k_a) \leftarrow \lceil (1+\alpha)^{k_a[brazo]} \rceil$
        $rondas\_restantes \leftarrow \tau(k_a) - k_a[brazo]$
        return $brazo$
    end if
end for
$t \leftarrow \sum conteos$
for cada $brazo$ en $\{0, \ldots, k-1\}$ **do**
    $\tau(k_a) \leftarrow \lceil (1+\alpha)^{k_a[brazo]} \rceil$
    $UCB2[brazo] \leftarrow valores[brazo] + \sqrt{\frac{(1+\alpha)\log((e \cdot t)/\tau(k_a))}{2\tau(k_a)}}$
end for
$brazo\_seleccionado \leftarrow \arg\max(UCB2)$
$\tau(k_a) \leftarrow \lceil (1+\alpha)^{k_a[brazo\_seleccionado]} \rceil$
$\tau(k_a + 1) \leftarrow \lceil (1+\alpha)^{k_a[brazo\_seleccionado]+1} \rceil$
$rondas\_restantes \leftarrow \tau(k_a + 1) - \tau(k_a)$
return $brazo\_seleccionado$

Algorithm 4 Selección de brazo con Softmax

Require: $k > 0$ (número de brazos), $\tau > 0$ (temperatura)
Ensure: Índice del brazo seleccionado
1: Inicializar $Q \leftarrow 0$ para cada brazo
2: Calcular los valores escalados: $valores\_escalados \leftarrow \frac{Q}{\tau}$
3: Estabilidad numérica: $max\_value \leftarrow$ máx$(valores_escalados)$
4: Calcular los valores exponenciales: $exp\_values \leftarrow \exp(valores\_escalados - max\_value)$
5: Calcular probabilidades: $p_i \leftarrow \frac{exp\_values[i]}{\sum exp\_values}$ para cada i
6: Seleccionar un brazo al azar $brazo \sim$ Distribución$(p_0, p_1, \ldots, p_{k-1})$
7: **return** $brazo$

Algorithm 5 Selección de brazo con Gradiente de Preferencias

Require: $k > 0$ (número de brazos), $\alpha > 0$ (tasa de aprendizaje)
Ensure: Índice del brazo seleccionado
1: Inicializar $H \leftarrow 0$ para cada brazo (preferencias iniciales)
2: Inicializar $R\_promedio \leftarrow 0$ (recompensa promedio)
3: Inicializar $t \leftarrow 0$ (contador de tiempo)
4: Estabilizar las preferencias: $H\_estable \leftarrow H -$ máx(H)
5: Calcular las probabilidades $\pi_t(a) \leftarrow \frac{\exp(H\_estable)}{\sum \exp(H\_estable)}$
6: Seleccionar un brazo $brazo \sim$ Distribución$(\pi_0, \pi_1, \ldots, \pi_{k-1})$
7: **return** $brazo$

Algorithm 6 Actualización del Gradiente de Preferencias

Require: $brazo\_elegido$ (índice del brazo seleccionado), $recompensa$ (recompensa recibida)
Ensure: Actualización de las preferencias H
1: Incrementar $t \leftarrow t + 1$
2: Actualizar el promedio de recompensa: $R\_promedio \leftarrow R\_promedio + \frac{(recompensa - R\_promedio)}{t}$
3: Estabilizar las preferencias: $H\_estable \leftarrow H -$ máx(H)
4: Calcular las probabilidades actuales: $\pi_t(a) \leftarrow \frac{\exp(H\_estable)}{\sum \exp(H\_estable)}$
5: Calcular la diferencia de recompensa: $d\_recom \leftarrow recompensa - R\_promedio$
6: **for** cada $brazo$ en $\{0, \ldots, k-1\}$ **do**
7: **if** $brazo == brazo\_elegido$ **then**
8: Actualizar $H[brazo] \leftarrow H[brazo] + \boldsymbol{\alpha} \cdot d\_recom \cdot (1 - \boldsymbol{\pi}_t(brazo))$
9: **else**
10: Actualizar $H[brazo] \leftarrow H[brazo] - \alpha \cdot d\_recom \cdot \pi_t(brazo)$
11: **end if**
12: **end for**

4. Evaluación/Experimentos

Es posible que para sus resultados use figuras y tablas como la Figura 1 o la Tabla 1. Si la tabla fuera demasiado grande use `sidewaystable` y póngala al final del documento.

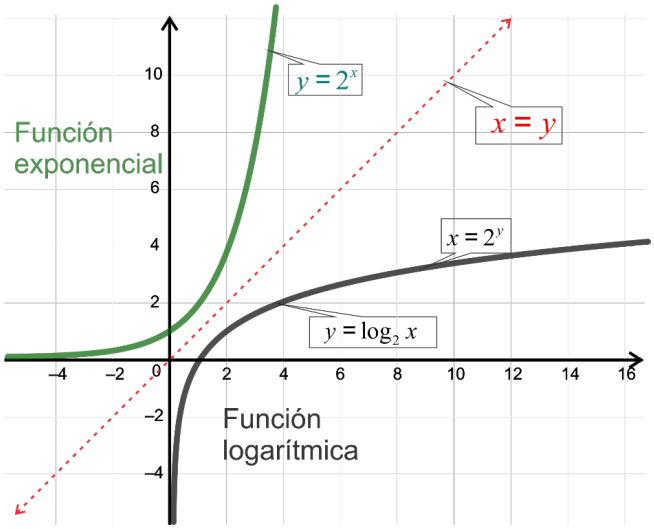

Figura 1: Esto es un ejemplo de figura

Tabla 1: Ejemplo de Tabla

Column 1	Column 2	Column 3	Column 4
row 1	data 1	data 2	data 3
row 2	data 4	data 5[1]	data 6
row 3	data 7	data 8	data 9[2]

[1] tablefootnote 1
[2] tablefootnote 2

5. Conclusiones

- Limitaciones del estudio y posibles mejoras futuras.
- Reflexión sobre la importancia del trabajo y su impacto en el campo del aprendizaje por refuerzo.
- Líneas *futuras* de estudio.

6. Repositorio

El repositorio en el que se realiza el trabajo es el siguiente: https://github.com/DeMiKe16/k_brazos_VLB.git